

REMARKS

Claims 1-6, 8-19, and 21-36 were pending. Claims 1-3, 12-14, 16, 25 and 28 have been amended and claims 37-42 have been added. Accordingly, claims 1-6, 8-19, and 21-42 are pending.

Allowed Claims

In the present Office Action, claims 4-6, 17-19 and 29-31 are allowed.

Claim Rejections

Claims 1-3, 8-16, 21-24, 26-28 and 33-36 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,691,768 (hereinafter “Civinlar”). Applicant respectfully traverses these rejections as submits each of the pending claims recite features neither taught nor suggested by the cited art.

Claim 1 recites a method which includes “receiving the multiple MPEG-encoded video streams; determining a value for a display position code corresponding to a display position of each slice of each of the MPEG-encoded video streams; modifying the value of the display position code of each slice of each of the received MPEG-encoded video streams as necessary; and interleaving each slice of each of the MPEG-encoded video streams as modified into a single composite video stream; wherein said display position code includes a first macroblock address increment variable length codeword having a first number of bits and wherein said modifying comprises: determining a second macroblock address increment variable length codeword that has a number of bits modulo 8 which is equal to said first number of bits modulo 8; and modifying the first macroblock address increment variable length codeword to be equal to the second macroblock address increment variable length codeword.” Applicant submits at least the above highlighted features are neither taught nor suggested by the cited art.

In contrast to claim 1, Civinlar teaches:

"In forming the four separate resolution bitstreams, in addition to the Slice Start Code renumbering that processor 108 must effect described above, the MAI associated with the first macroblock in each slice in each stream is also likely to require changing to properly reference each slice to beginning of its new lower resolution picture. Thus, for example, in forming the 320X240 (field 1) bitstream, the MAI in each slice having SSC=2 in the 320X240 (field 1) picture is decreased by 20, the MAI in each slice having SSC=3 in this same picture is decreased by 40, the MAI in each slice having SSC=4 is decreased by 60, etc., so that the resultant MAIs in this bitstream are properly referenced to the 320X240 resolution size, which contains only 300 macroblocks. The MAIs in the other resolution bitstreams are similarly renumbered in accordance with their position in frame buffer 107 and their particular resolutions.

As previously discussed, the location of the MAI in each slice is readily locatable since it is very close to the Slice Start Code. Accordingly, it can be located and renumbered without decoding the slice bitstream. This MAI is variable length coded and therefore not byte aligned. Frequently, therefore, renumbering of this address necessitates bit level shifts for the rest of the slice to ensure that all succeeding slices in the demultiplexed bitstream remain byte aligned. Thus, binary '0's are added to the end of the slice's bitstream, where needed, to compensate for renumbering the slice's MAI." (Civinlar, col. 8, line 66 – col. 9, line 25).

The above teachings of Civinlar clearly show that Civinlar teaches renumbering the MAI may "frequently" necessitate "bit level shifts for the rest of the slice." In order to compensate for these bit level shifts, Civinlar teaches adding binary '0's to the end of the slice bitstream. In contrast to Civinlar, claim 1 recites a method quite different from that taught above. As noted in Applicant's Description:

"By modifying the MBAI VLC to another MBAI VLC having the same number of bits modulo 8, the resulting MBAI VLC will have the same bit position within a byte as the original MBAI VLC, i. e., the bit alignment within a byte would be unchanged. In other words, by having the MBAI VLC maintain the same bit alignment within a byte obviates the need to rotate all bits in the slice following the MBAI VLC bits, such as in a shift register. Avoiding such shifting or rotation of the remaining bits in the slice avoids a relatively expensive process." (Description page 30, line 21 – page 31, line 4).

Therefore, Civanlar teaches (1) modifying the MAI, (2) shifting slice bits, and (3) padding the slice with 0s to compensate for the shifting. Consequently, the recited features of claim 1 of “determining a second macroblock address increment variable length codeword that has a number of bits modulo 8 which is equal to said first number of bits modulo 8; and modifying the first macroblock address increment variable length codeword to be equal to the second macroblock address increment variable length codeword” are wholly absent from the cited art. Further, rather than providing any teaching or suggestion concerning these recited features, the cited art merely serves as a prior art example of a method and mechanism which requires shifting of slice bits.

In addition, Applicant respectfully traverses the Official Notice related to claim 1. Applicant first notes the Official Notice makes reference to the last 8 bits of the slice start code which indicate the vertical position of the slice. However, in contrast to the slice start code, the features of claim 1 discussed above relate to the macroblock layer header and the macroblock address increment in particular. Consequently, the Official Notice does not directly address the above recited features. Nevertheless, in view of the above discussion, Applicant submits the above highlighted features are not obvious and were clearly not contemplated by the cited art. However, should the Examiner believe that the above highlighted features are obvious, Applicant respectfully requests the Examiner provide documentary proof in support of the assertion.

Accordingly, Applicant submits claim 1 is in condition for allowance. Further, because each of claim 12 and 25 include features similar to that of claim 1, each of claims 12 and 25 are believed allowable as well.

With respect to claims 9, 22 and 34, Applicant submits each of these claims recite features neither taught nor suggested by the cited art. In the Office Action it is suggested that claims 9, 22 and 34 recite the same subject matter as claims 1, 12 and 25. However, Applicant submits this is not the case. Further, the Office Action suggests that Civanlar discloses adding stuffing codes. However, neither is this the case. As already discussed, Civanlar merely teaches renumbering a MAI, shifting slice bits, and padding the slice bitstream with 0s. There is no mention whatsoever in Civanlar of using macroblock stuffing codes. Further, as Civanlar

explicitly teaches shifting slice bits, Civinlar does not contemplate the use of macroblock stuffing codes in the manner as recited in the claims of the present application.

Claim 9 recites a method “wherein said MPEG-encoded video streams are MPEG-1 encoded video streams, and wherein said display position code includes a macroblock address increment (MBAI) codeword, wherein said modifying the display position code of each slice of each of the MPEG-1 encoded video streams as necessary includes selectively adding a number of MBAI_stuffing codes, said number of MBAI_stuffing codes ranging from 0 to 7.” Therefore, modifying the display position code *includes* adding between 0 and 7 stuffing codes. Nothing in Civinlar teaches or suggests these features. To further assist in appreciating the inventiveness of the recited method, the following portion of the present application’s description is provided:

“The system and method of the present invention may also be utilized in the context of MPEG-1 compression. In addition to the MBAI code, MPEG-1 compression provides an addition MBAI_stuffing code. The MBAI-stuffing code has 11 bits, a prime number. Thus, the MBAI code can be bit-aligned to any bit position within a byte by adding a selective number of the MBAI_stuffing code. It is noted that in the case of MPEG-1 encoded video streams, an additional requirement is that all slices start and end on the same row.

FIG. 8 is a table listing the number of MPEG-1 MBAI_stuffing codes, the corresponding number of bits, and the corresponding number of bits modulo 8. As shown, adding 0,1,2,3,4,5,6, or 7 MBAI-stuffing codes results in a bit-alignment shift of 0, 3,6,1,4,7,2, or 5 bits. Thus, the MBAI code can be bit-aligned to any bit position within a byte by adding between 0 and 7 MBAI_stuffing codes.

FIG. 9 is flow diagram illustrating a process 400 of an MPEG-1 interactive decoder of the present invention in processing a plurality of video streams for simultaneous transmission and rendering. At step 402, the interactive decoder determines the destination MBAI VLC and/or the corresponding increment value of the slices for each video stream to be repositioned. At step 404, the interactive decoder determines the number of MBAI stuffing codes to be added to the MBAI such that, after the MBAI is modified, the resulting code is bit-aligned to the desired bit position within a byte. At step 406, the interactive decoder modifies the MBAI value of each slice of the video stream to be repositioned such that the video slice would be repositioned at the desired location. In addition, the interactive decoder inserts the predetermined number of MBAI stuffing codes into the slice so as to maintain the same bit-

alignment of the MBAI within a byte.” (Description, page 33, line 15 – page 34, line 15).

Therefore, in the case of an MPEG-1 stream, an alternative inventive approach may be utilized which maintains bit alignment within a byte and obviates the need for bit shifting. None of the cited art teaches, suggests, or even contemplates such an approach. Consequently, Applicant also respectfully traverses the Official Notice taken with respect to claims 9, 22 and 34. Should the examiner wish to maintain the rejection, Applicant requests documentary proof in support of the rejection.

Still further, Applicant submits not all of the features of claim 14 are taught or suggested by the cited art. Claim 14 recites a system “wherein in modifying a display position code for a given slice, said interactive decoder is further adapted to: determine a modified display position code which maintains a same bit-alignment within a byte as an original display position code of said given slice; and modify the value of the original display position code to be equal to the modified display position code.” As already discussed, the cited art includes no such teaching or suggestion of these features. In contrast, the cited art explicitly teaches renumbering a display position code followed by shifting and slice padding to account for the fact that the renumbered display position code does not maintain bit-alignment as recited.

In view of the above discussion, Applicant submits each of the pending independent claims, 1, 9, 12, 14, 22, 25, and 34 are patentably distinct over the cited art and are believed in condition for allowance. Further, because each of the dependent claims include at least the features of the independent claims upon which they depend, each of the dependent claims are believed allowable as well.

In addition, Applicant notes that the dependent claims include additional features neither taught nor suggested by the cited art. Selected examples are discussed below.

Claim 2 recites the additional features “wherein said modifying the value of the display position code is performed at an end user location.” Applicant submits this feature is neither taught

nor suggested by the cited art. Civanlar, for example, explicitly teaches performing modifications prior to transmitting the video stream. In addition, Civanlar goes on to teach:

“Thus, by transmitting to the user only that bitstream associated with the size of the image requested to be displayed rather than a full resolution bitstream, substantial bandwidth can be saved as can the processing power to decode the full-resolution bitstream and to scale the resulting video to the desired less than full resolution image size.” (Civanlar,).

Consequently, were the modifications described in Civanlar to be performed at the end user site, the above described advantages (i.e., reduced bandwidth and processing power) would be lost. Rasmussen similarly describes a system wherein “[t]he video communications system minimizes the use of bandwidth.” (Rasmussen, Abstract). Accordingly, the cited art neither teaches nor suggests this feature.

In addition, claim 3 recites the additional features “wherein the multiple MPEG-encoded video streams are selected by said end user from a list of available video streams.” These additional features are described in the Summary of the Invention, as well as other portions of the description. Applicant submits these features are wholly absent from the cited art.

Still further, claim 10 recites “wherein said number of MBAI_stuffing codes is determined such that the macroblock address increment codeword maintains bit-alignment of the display position code within a byte” which is absent from the cited art.

Similarly, claim 11 recites “wherein said macroblock address increment codeword has a first number of bits and wherein said modifying the display position code of each slice of each of the MPEG-encoded video streams to be modified results in a modified macroblock address increment codeword and a predetermined number of MBAI_stuffing codes, the modified macroblock address increment codeword and the predetermined number of MBAI_stuffing codes combine to having a modified number of bits, said modified number of bits modulo 8 is equal to said first number of bits modulo 8” which is absent from the cited art.

Applicant believes the application is now in condition for allowance. Should the examiner have any questions or comments, the below signed representative would be more than happy to discuss them via phone at (512) 853-8866.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5266-09800/RDR.

Respectfully submitted,

Rory D. Rankin
Reg. No. 47,884
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin,
Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: March 29, 2004